



# LIGHTS! CAMERA! ACTION!™

## User's Guide



**David Barrett • John Sievers**

ISBN: 0-915595-30-3



PaperDisk Publishing  
Los Angeles, CA

First Edition

First Printing

Copyright © 1988 PaperDisk Publishing

Reproduction or publication of the content in any manner, without the express written consent of the publisher, is prohibited. No liability is assumed with respect to the use of the information herein.

Library of Congress Cataloging in Publication Data  
Barrett, David  
Sievers, John

ISBN 0-915595-30-3

Book Design by PaperDisk Publishing

Lights! Camera! Action!, ShowLCA Copyright © 1988 Sparta, Inc. VideoTitler Copyright © 1987 Sparta, Inc. Aegis ArtPak Copyright © 1987 Aegis Development, Inc. SuperGen is a trademark of Digital Creations, Inc. Deluxe Paint is a trademark of Electronic Arts. Amiga, Workbench, and Kickstart are trademarks of Commodore, Inc. Lights! Camera! Action!, VideoSEG, VideoTitler, and ShowLCA are trademarks of Aegis Development, Inc.

## **ACKNOWLEDGEMENTS**

Thanks to all of the creative people who have developed or published graphic programs for the Amiga. *Lights! Camera! Action!* has been designed to bring all of those talents together.

*A special thanks to Aegis staff John Sievers, Scott Oransky, Elizabeth Vanture, Jennifer Capp, and Lights! Camera! Action! author Gary Bonham.*

# **CONTENTS**

---

**INTRODUCTION .....** vii

**1**

<b>GETTING STARTED .....</b>	1
What You Need to Run <i>Lights! Camera! Action!</i> ..	1
Loading <i>Lights! Camera! Action!</i> .....	1
Using the Mouse .....	3

**2**

<b>A QUICK TOUR .....</b>	5
The Menu Bar .....	6
Playing a Sample Script .....	6
Creating a Sample Script .....	8

**3**

<b>CREATING SCRIPTS .....</b>	15
The Video Script Requestor .....	15
The Frame Specification Requestor .....	18
Saving a Script .....	41
Playing Scripts .....	42

**4**

<b>DISPLAY OPTIONS .....</b>	45
NTSC/PAL.....	45
Lo-res.....	47
Video-res.....	47
Med-res.....	48
Hi-res.....	48
Interlace.....	48
HalfBrite.....	48
Hold and Modify.....	49
Cycling .....	49
The Attention Requestor.....	49

**APPENDICES**

<b>A THE STORAGE REQUESTOR.....</b>	51
<b>B USING GrabANIM .....</b>	55
<b>INDEX .....</b>	57

**INTRODUCTION**

Welcome to the exciting world of desktop presentation. This User's Guide is your reference to *Lights! Camera! Action!*, a versatile and powerful presentation program for the Amiga family of computers. This introduction will give you some background on desktop presentations, provide an overview of *Lights! Camera! Action!*, and give you some tips on how to get the most out of this User's Guide.

**What is Desktop Presentation?**

Why desktop presentation? The answer is simple. We live in a visual society. Everything we do these days is based on visuals. Television is our number-one source for information. We use video tape to communicate almost anything you can think of... news, video dating services, real estate sales, home shopping, the list goes on and on.

The people who react to these visuals don't come from one type of background either. Desktop presentation isn't just for accountants, or architects, or engineers. Everyone in our society reacts to visual stimulation. In fact, they always have.

If a picture is worth a thousand words, then computers that generate graphic images must be worth volumes. Any presentation that has a hope of winning acceptance includes charts, diagrams, tables, and graphs. The presenter will

highlight the key points of his concept and use the power of his visual aides to convince his audience that his point of view is correct.

That visual form of communication is what desktop presentation is all about. It isn't easy to explain a spreadsheet of numbers to someone not familiar with financial statements. If you show them a graph of your company's growth - *a picture of those numbers*, the message becomes crystal clear.

What about sound? If you can give your images an aural backdrop, you can add 100% to the influence your presentation will have on the audience watching it. Sound effects, people talking, and music are critical to the desktop presentation environment.

## About Lights! Camera! Action!

*Lights! Camera! Action!* has been designed for anyone who wants to combine pictures, animations, music and sounds into a desktop presentation using the Amiga computer or video tape. The program can work in conjunction with other programs, including *Aegis VideoTitler*, *VideoScape 3D*, *Sonix*, *AudioMaster*, and *Images*.

*Lights! Camera! Action!* allows you to combine a series of images or ANIM-style animations into a continuous presentation. A completed presentation is called a script and works very much like a movie script. Each page of the script includes information on the type of image, how long it is shown, what sounds (if any) are included, and what type of transition will be used to move to the next image or animation.

A presentation module called *ShowLCA* allows you to build a disk with your presentation and distribute it without the *Lights! Camera! Action!* program. As an alternative, you may record your presentation onto video tape for easy distribution without an Amiga computer.

*NOTE: ShowLCA is not a public domain program, but may be freely distributed by registered owners of Lights! Camera! Action!*

Also included is *GrabANIM* - a utility program which lets you create ANIM-style animations by "grabbing" a series of screens from virtually any Amiga program.

## Key Features of Lights! Camera! Action!

*Lights! Camera! Action!* allows for pictures, animation, and sound to be organized into a complete presentation. Some of *Lights! Camera! Action!*'s key features include:

- **Supports the Aegis/Sparta ANIM format:** Any animation created using the Aegis/Sparta ANIM format can be included in a presentation. Programs that currently support the ANIM format include *Aegis VideoTitler*, *VideoScape 3D*, and *The Director*™.
- **Multiple resolutions with overscan:** *Lights! Camera! Action!* will work in each of the Amiga's display resolutions with overscan. The program will also support interlace in any resolution. Overscan is very important if you are considering recording any of your presentations onto video tape.

*Note: Make sure that the ANIM's and pictures you assemble in a script were all created in the same resolution; otherwise, you'll get either truncated or distorted displays.*

- **Scene transitions:** *Lights! Camera! Action!* allows you to choose from over 40 different transitions when moving from one image or animation to the next. These transitions will give your presentation that added sparkle.
- **Sound and music:** You can add sound to your presentations as well as sampled sounds. *Sonix*-style SMUS scores can be

## LIGHTS! CAMERA! ACTION!

created in *Sonix* and then attached to a presentation. You can trigger instruments and IFF sampled sounds (saved in *Sonix* format) to play during any image display or on specific frames of an animation.

- **Supports color cycling:** If you want to create an animation effect using color cycling in a paint program, *Lights! Camera! Action!* will allow the cycling to function while the image is displayed.
- **Supports HAM:** The Hold and Modify mode of the Amiga is supported. This means you can display images with up to 4096 colors. You can also run ANIM's created with HAM images. HAM is useful for digitized images such as color photographs or frames from a video tape. It's important to note that HAM images use quite a bit more memory due to their highly complex color tables. If you use HAM images extensively, you may want to consider using a hard disk drive in order to develop presentations of any significant length.

# CHAPTER

# 1

# GETTING STARTED

*Lights! Camera! Action!* is found on the main program disk.

## What You Need to Run Lights! Camera! Action!

*Lights! Camera! Action!* will work with any Amiga computer and 1MB of memory (RAM) and two disk drives. *Workbench* version 1.2 (or greater) is required.

*Note: For optimum performance, you may want to consider using 2 MB of RAM or more. Certain ANIM-style animations or complex images may require more than 1MB of RAM even though the *Lights! Camera! Action!* program does not.*

## Loading Lights! Camera! Action!

To start *Lights! Camera! Action!* from the Workbench:

- Make sure the Amiga and monitor are turned on.

- Insert the main program disk in the internal drive and the data disk in the external drive. Make sure the disk labels face up and the metal slide that covers the disk is inserted into the drive first.

*Note: The Amiga 1000 requires a disk called KickStart before the Lights! Camera! Action! disk can be used. All Amiga models produced since the 1000 do not require Kickstart.*

- Double-click with the Selection (left) mouse button on the disk icon. A window will appear containing more icons.
- Double-click with the Selection (left) mouse button on the *Lights! Camera! Action!* icon. After a half minute or so, the *Lights! Camera! Action!* program will appear on the screen.

You can also run *Lights! Camera! Action!* from the CLI (Command Line Interpreter, described in the handbook accompanying your Amiga). For additional information on the use of the workbench and CLI, please refer to the manuals supplied with your Amiga or one of the popular books about the Amiga, such as *The AmigaDOS Manual* published by Bantam Books.

To start *Lights! Camera! Action!* from the CLI:

- At the CLI prompt, type LCA.

You may run more than one program on the Amiga at a time. This is called "multi-tasking."

To access the Workbench or CLI after starting *Lights! Camera! Action!*:

- Hold down the left Amiga key and press the N key. You will be returned to the Workbench screen (or the CLI if you started from there).
- Hold down the left Amiga key and press the M key to return to the *Lights! Camera! Action!* screen.

## Using the Mouse

The Amiga Mouse performs three general tasks: "choosing," "selecting," and "dragging."

### Choosing

The right mouse button is called the Menu button. You use it to choose items from the Menu Bar. You can see the Menu Bar in *Lights! Camera! Action!* when you press the Menu button. To choose a menu item, put the cursor on a menu heading and move it down until the item you want is highlighted. Release the Menu button and that item is chosen.

### Selecting

The left mouse button is called the Selection button. One of its uses is to "select" items such as buttons and arrows. For example, to select a button, put the cursor on it and click the Selection button.

### Dragging

The Selection button is also used to "drag" items, such as icons (symbols) on the Workbench. To move an item to a new place, put the cursor on it and move the item while holding down the Selection button.

## CHAPTER 2 | A QUICK TOUR

When *Lights! Camera! Action!* first loads you are presented with a blank screen and the **About** message. This message provides you with the basic information on the program. Underneath the **About** message is the free memory display. The amount of free memory is important. If you think there is a problem with your computer, one of the first things to do is check the **About** message for the amount of free memory.

To close the window, press any key or mouse button.

*Lights! Camera! Action!* starts in the "overscan" mode. Overscan is a display mode where the image extends beyond the edge of the display monitor. For this reason the menu bar cannot be seen until you hold down the Menu (right) mouse button.

## The Menu Bar

---

### Project

Commands for loading and saving scripts are located under **Project**. The **Project** menu also contains commands to bring up the **About** message for displaying the amount of free memory, the **Sleep** command for multi-tasking with other programs, and the **Quit** function.

### Video

Presentation scripts can be assembled or modified using the **Edit** selection. Once you've completed a presentation, you can select from a number of **Play** options.

### Display

The selections under this menu choose the parameters for the presentation being created. The resolution, whether it has interlace, halfbright, and color cycling are all chosen here. A checkmark will indicate which options are selected.

## Playing a Sample Script

---

The best way to get a feel for what *Lights! Camera! Action!* does is to see a completed presentation. There is a sample script on the data disk that contains an animation and several still pictures which use different transitions.

To load and play a sample script:

- Choose **Open Script** from the **Project** menu. A storage requestor will appear (described in Appendix A, "The Storage Requestor"). The **drawer** field should read **df0:**. Click in the

field, press the **BACKSPACE** and/or **DEL** keys to erase what's there, and type in the following: **df1:Scripts/**. Press **RETURN**.

- From the list of scripts, click on **Sample.Script** with the **Selection** (left) mouse button (this action is called "selecting"). The name will appear in the **file** field.
- Select **OK**.
- Press the **Menu** (right) mouse button to gain access to the pull-down menus of *Lights! Camera! Action!*
- While holding down the **Menu** button, position the cursor over the **Video** menu. You'll see several choices appear.
- Move the cursor down the list until it is positioned over **Play Once** and release the mouse button. This action is called "choosing."

The presentation will play through one time and stop. To see it again choose the **Play Once** command again or **Play Loop** if you want it to repeat until you press the **Menu** (right) mouse button.

As an alternative, you can make selections using certain keyboard sequences. For example, to see the presentation played once, type the **Amiga** key (to the right of the space bar) and the letter **P** together. If you've already played the script once, you'll have to click the **Menu** (right) mouse button before you can use the keyboard sequences. As you move through the different pull-down menus, you'll see a number of keyboard sequences to choose from.

## Creating a Sample Script

Now that you've seen the results of a completed presentation, let's build a simple script from some of the sample information provided on the *Lights! Camera! Action!* data disk.

This presentation will have an opening or title screen, an picture, and an ANIM animation. We'll add a musical backdrop to the presentation with a *Sonix* score.

### Creating a Frame:

- Use the right button of the Amiga mouse to gain access to the pull down menus of *Lights! Camera! Action!*
- Start with the **Project** menu and choose **New**. An attention requestor will appear asking if you really want to start a new script. Select (use the left mouse button) **Yes**.
- Now choose **Edit** from the **Video** menu. The **Video Script** requestor should appear.
- You'll notice that frame number 0 reads, **\*Start black\***. This is the entry frame of all presentations and doesn't need to be adjusted.
- Select the **\*End Black\*** edit field (using the selection (left) mouse button).
- Next, select the **Insert** button. The new line will be inserted above the currently selected line. Until you assign a picture or animation to the frame, it will be called **\*Blank\***.
- Select **Edit** from the lower center of the window

The Frame Specification (**Frame Specs**) requestor will appear. This is where you can select the type of frame you want (Eg.

picture, sound, etc...) and what will happen with it. To begin with, we'll select a musical score to play during the presentation.

### Scores and their instruments:

Before using the **Load Score** command in a script, you have to tell the *Lights! Camera! Action!* program where it can find the instruments required by that score. This lets you organize your disks so that the scores and instruments can be in separate places (even in RAM:), but it's a good idea to put both scores and their instruments in the same directory. In order to tell the program where the instruments are located, we "assign" the instrument directory using the following steps.

### Assigning the Instrument Directory:

- In the **Video Script** window, create a frame in your script which comes *before* the **Load Score** frame.
- Select that frame and then select **Edit**.
- Select **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list of frame options and choose, **Load Sound**.
- Enter a 0 in the **Buf** field. We're using buffer 0 as a dummy buffer just to set the path to our instruments.
- Click in the file edit area using the left mouse button. A requestor will appear allowing for the file path to be changed. (See Appendix A: The Storage Requestor.)
- Select any instrument in the directory where your instruments are located. (In the case of the data disk that came with your program, the instruments are in the **Music** directory with the scores.) Since we are only setting the path to be used, any instrument can be used.

- Click **OK** in the requestor and select the gadget in the upper left corner of the **Frame Specs** window.

#### Loading a score:

- Select **Blank** in the upper left corner of the **Frame Specs** window and hold the button down. A "pop-up" menu will appear.
- Move down the list of frame options until **Load Score** is highlighted and choose it by releasing the selection button.
- Click the left button of the mouse in the **Buf** edit field, use the **BACKSPACE** and/or **DEL** keys to erase what's there, and type a **1**. This will be the buffer, which is a chunk of the computer's memory, you use for the score. You need to load the score into a buffer before you can play it. You can use any number from 0 to 99 for each buffer that you use.
- Select **Blank** in the **File** field and a requestor will appear allowing you to load the sound file and the path for that file into the selected buffer.
- Type in **DF1:Music/Sample.Score** in the file edit field. (Or see Appendix A for quicker ways to move around in the Storage Requestor.)
- Select **OK** from the lower left corner of the requestor.
- Select the close gadget in the upper left corner of the **Frame Specs** window.

#### Playing a score:

Once you've loaded the score, we'll want to begin playing it right away, even as new images are loaded into your Amiga's memory.

- Return to the **Video Script** requestor and create another frame (select **End Black** and then **Insert**).

#### Select Edit.

- Select **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list of frame options and choose **Play Score/Sound**. You'll see that the **Time**, **Buf** and **Score/Sound** fields are the only options available with this frame selection.
- Normally, the next step is to position the cursor in the **Time** edit field at the top of the requestor and enter the number of seconds you want the score to play. In this case, we will want the score to play until the end of the presentation. Therefore, leave the **Time** edit field as it is, with **0** entered.
- In the **Buf** edit field, enter the number of the buffer you assigned to the score when you loaded it from disk (number **1**). There are a number of options available when playing a score or sound. You will not use the **Track** option for scores as they have already been determined when the score was created in *Sonix*. Other options will be covered in the next chapter.

#### Displaying a still image:

- Return to the **Video Script** requestor and create another frame.
- Select **Edit**.
- Select **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list of frame options and choose **Show still**.
- Enter the number of seconds you wish the image to be displayed on the screen in the **Time** edit field. For this example, type **10**.

The image will display for 10 seconds when it is loaded from disk.

*Note: To make the script run more smoothly and without pauses while pictures are loaded from disk, you could have loaded the picture into a buffer first, as you did with the score. For this learning example, we'll just stick to loading the picture from disk.*

- Select **Blank** in the **File** field.
- Type **DF1: PICS** in the **drawer** edit field and press the **RETURN** key. You will see a list of the available images.
- Type **Title1.Pic** in the **file** edit field or scroll through the list of pictures and select it. This is the name of the image you want displayed.
- Select **OK**.
- Now you'll need to select the type of transition you want. In the lower area of the **Frame Specs** requestor, you'll see a variety of options. For this example, select **Dissolve**. That button will become highlighted.
- Return to the **Video Script** requestor by selecting the close gadget in the upper left corner of the **Frame Specs** Requestor.

#### Loading an animation:

- You will need 3 frames, so after you select **End Blank**, press the **Insert** button 3 times to create the blank frames.
- Select the **Botm** button to scroll down to the bottom of the list of frames so that you can see your new frames.
- Select the first frame and then select the **Edit** button.

- In the **Frame Specification** requestor select the **Blank** button and choose **Load Buff** from the list of frame options.
- Change the **Buf** field to **2**.
- Select **Blank** in the **File** field.
- Type **DF1: ANIMS** in the **drawer** field.
- Select the **File** field and type in **Sample.ANIM**.
- Select **OK**. The action just created will load the animation into memory but not play it.
- Close the **Frame Specs** requestor.
- Select the next **Blank** frame and then the **Edit** button. Before we play the animation, we are going to bring up the first frame of the animation as a picture (**Show Still**). This way, we can have a special effects transition (**Dissolve**) into the animation.
- Select the **Blank** button and choose **Show Still**.
- Change the **Buf** field to **2**.
- Now you'll need to select a transition. For this example, select **Dissolve**.
- Enter the number of seconds to pause before beginning the animation. For this animation, type in **2** in the **Time** edit field.
- Close the **Frame Specs** requestor.
- Select the final **Blank** frame and then the **Edit** button.

*Note: If you can't see the next blank field, choose the **Next** button in the bottom left of the **Video Script** window to scroll down the list of frames.*

- Select the **Blank** button and choose **Play ANIM**.
- Change the value in the **Buf** field to 2.
- Close the requestor.
- Choose **Play Once** from the **Video** menu. When played, the script just created will pause as it loads all of the files and then perform the chosen transition into the first frame of the animation before playing it.

## CHAPTER

# 3 CREATING SCRIPTS

As you have seen, presentations are primarily built with the **Video Script** and **Frame Specs** requestors. The **Video Script** requestor is a scrolling list of the pictures, ANIM's, music and other events in the show. You reach it through the **Edit** command of the **Video** menu. The **Frame Specs** requestor is accessed from there for each frame of the script.

### The Video Script Requestor

Scripts are built with the **Video Script** requestor reached by choosing the **Edit** command from the **Video** menu. The requestor is divided into two basic areas: the frame list and the buttons controlling the list (see Figure 1).

The list has four columns. Starting on the left is the frame number. Next is **Time** which shows how long the picture will be displayed in seconds.

The next column, **Frame**, is the name of the picture or ANIM being displayed. For either case the filename is used. This

column may also hold other script commands such as Loop Point or Pause.

To the right is the **Action** column. This describes the action or transition which will occur between frames. The rows of this column are offset to show where they occur in the sequence. The transition before a frame is displayed is chosen when the details of the frame are specified. This is done in the **Frame Specification** requestor described later in this chapter.

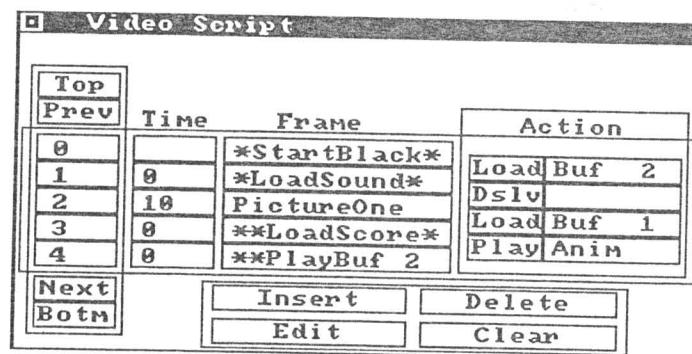


Figure 1: The Video Script Requestor.

### Scrolling the List

There are eight buttons controlling the frame list. Above and below the frame number column are four buttons used for scrolling the list. **Top** moves you to the top of the list while **Prev** scrolls the list up one row. **Botm** displays the bottom of the list and **Next** scrolls the list down one row.

### Insert

Below the list are four buttons used for editing the list. **Insert** will add a new line to the list above the currently selected line. All of the rows are shifted down one to make room for the new line and the frame numbers are increased by one.

#### To add a line to a script:

- In the **Frame** column, select the line below where you want to insert a line. It will be highlighted to show it's selected.
- Select the **Insert** button. The new line will be inserted above the currently selected line.
- Select **Edit** from the lower center of the window

### Delete

The **Delete** button removes the currently selected line from the script. The associated transition is also removed.

#### To remove a line from the script:

- In the **Frame** column, select the line to remove. A blue highlight bar will appear around it when it is selected.
- Select the **Delete** button.

### Clear

Selecting **Clear** will erase the entire script so that you may start over.

*Note: There is no UNDO in Lights! Camera! Action!, so make sure you delete the right line. If you don't you'll need to recreate it.*

## Edit

The **Edit** button displays the **Frame Specs** requestor for the currently selected line in the script. The requestor is described in detail below. The information displayed in the frame list such as time, action, transition, and frame name are all entered in this requestor.

To enter or edit the information on a line in the list:

- In the Frame column, select the line to edit.
- Select the **Edit** button. The **Frame Specification** requestor will appear.

## The Frame Specification Requestor

This requestor has two main areas for defining a frame in the script (see Figure 2). The top half is dedicated to the type of action such as loading a picture or score or playing an animation, and the bottom half describes the type of transition that occurs before the action.

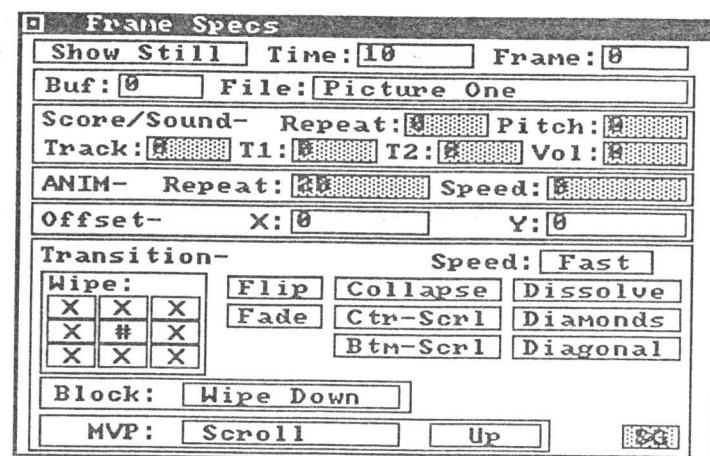


Figure 2: The Frame Specification Requestor

The **Frame Specs** requestor includes all of the instructions required to build a sequence in a presentation. The **Frame Specs** requestor uses "pop-up" menus to help speed up the selection process. For example, place the cursor on top of the button that reads **Blank** and press the left mouse button. A number of options will appear. These are the various functions available under the **Blank** button. For the moment, let go of the mouse button. You'll learn about the specific options in a few minutes.

## Buffers

A consideration in preparing a presentation is the amount of time it takes to load a picture or animation. Waiting for a picture to load can hinder you in making a smooth presentation.

*Lights! Camera! Action!* provides a solution to this problem. Both pictures and ANIM's can be loaded into memory prior to

being shown. The slot in memory they are loaded into is called a buffer. *Lights! Camera! Action!* has 100 (0-99) possible buffers. The numbers 0 through 99 are the names you give to the buffers for reference purposes. The number of actual buffers you may have is limited by the amount of memory in your machine.

*Note: When assigning buffer numbers, be careful not to use a buffer already allocated to another image, animation, score, or sound.*

### Blank Button

Select the button in the upper left corner that reads **Blank**. A pop-up menu will appear with a number of options. These options will indicate what happens during the sequence you are editing. The options work as follows:

### Blank

*Note: The use of individual buttons will be described in the order they appear in the menus. When using *Lights! Camera! Action!*, it may be necessary to use the buttons in a particular sequence.*

Using this action will cause the screen to blank to color zero for the amount of time specified in the **Time** field.

### Color Bars

Selecting color bars will place a series of colored vertical bars on the screen. These colors are based on the bars used in video production.

### Load Buff

This action loads either an IFF picture or an animation into memory for playback later in the show. The **File** field must

contain the name and location of the file. The **Buffer** field holds the number of the buffer to load the picture into.

The number of buffers you can have active depends on how much memory you have. The IFF files are stored in memory in compressed format, allowing you to load more than you could normally.

### Append Buff

This option is designed to support multi-file ANIMs. A large ANIM can be created with **GrabANIM** which will not fit onto a single floppy disk. By splitting the ANIM into smaller segments and using the **Append Buff** option, you can get *Lights! Camera! Action!* to combine the files into one large, continuous animation.

As an example, let's suppose that you have used **GrabANIM** to create an animation using three large ANIM files spread over three floppy disks with a total of 300 frames. The first as frames 1-100, the second as 99-200 and the third as 199-300. As in repeating ANIMs, it is required that the first two frames of a subsequent file match the last two frames of the preceding file exactly. Then the first file is loaded as usual into a buffer. In separate frames, the next two files are loaded into the same buffer number using **Append Buff**. It will ignore the first two frames of the file and simply add the rest of the frames to the existing data in the buffer. Then using the **Play ANIM** command with the proper buffer number, you can play back your big ANIM, even though it was recorded in sections on multiple floppy disks.

### Form Buffer

*Lights! Camera! Action!* normally stores pictures in buffers in a compressed format. This conserves memory but it also means that the program must take some time to decompress the picture before it can be displayed on the screen. **Form Buffer** allows

you to decompress a picture ahead of time that you have already loaded into a buffer. This will let you "flip" between pictures much more quickly.

- In separate frames do the following:
- Load a picture file into a buffer using the **Load Buff** command in the **Frame Specs** requestor and enter a Buf number.
- Use the **Form Buffer** command and specify the same buffer number.
- Choose the **Show Still** command and specify the same buffer number again.

### Clear Buff

The clear buffer command erases the contents of the specified buffer and frees the memory for other uses. The Buf field must contain the number of the buffer to erase.

### Load Score

You may select **Load Score** to add a musical soundtrack to your presentation. Pictures and animations can be loaded into a buffer or loaded straight from disk during a script. Scores, on the other hand, must be loaded into a buffer first. Thus we have the **Load Score** command which must be used before we can use the **Play Score/Sound** command. Currently, soundtracks must use the SMUS format as generated by *Sonix*.

### Scores And Their Instruments:

Before using the **Load Score** command in a script, you have to tell the *Lights! Camera! Action!* program where it can find the instruments required by that score. This lets you organize your disks so that the scores and instruments can be in separate places (even in RAM:), but it's a good idea to put both scores and their

instruments in the same directory. In order to tell the program where the instruments are located, we "assign" the instrument directory using the following steps.

#### Assigning the Instrument Directory:

- In the **Video Script** window, create a frame in your script which comes *before* the **Load Score** frame.
- Select that frame and then select **Edit**.
- Select **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list of frame options and choose, **Load Sound**.
- Enter a 0 in the **Buf** field. We're using buffer 0 as a dummy buffer just to set the path to our instruments.
- Click in the file edit area using the left mouse button. A requestor will appear allowing for the file path to be changed. (See Appendix A: The Storage Requestor.)
- Select any instrument in the directory where your instruments are located. (In the case of the data disk that came with your program, the instruments are in the **Music** directory with the scores.) Since we are only setting the path to be used, any instrument can be used.
- Click **OK** in the requestor and select the gadget in the upper left corner of the **Frame Specs** window.

#### Loading a score:

- Click the mouse on **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list and select **Load Score**.

*Note: Only the buffer number and file name can be edited with the Load Score selection. You must load a score into a buffer before it can be played.*

- Click the left button of the mouse in the buffer edit field and chose a buffer to load the score. You may choose from 0-99 as the buffer number. Be careful not to use a buffer already allocated to another image, animation, score, or sound.
- A requestor will appear allowing you to load the score file.
- Select OK and return to the Frame Specs window.

### Load Sound

Sampled sounds and *Sonix* instruments can be loaded into *Lights! Camera! Action!* to highlight a specific image or to compliment an animation. *Lights! Camera! Action!* will read IFF sampled sounds such as those created with *AudioMaster* (as long as they have been saved as *Sonix* instrument files). The procedure for loading sounds is the same as for scores, i.e. you must load the sound into a buffer before you use it.

### Show Still

This option displays an IFF picture for the amount of time specified in the **Time** field. The picture may be displayed directly off disk or displayed from a buffer. If the picture is to be read from disk and displayed in one action, the **File** field must contain the name and location of the file. The first frame of an ANIM file can also be read and displayed with this command.

**Show Still** is also used to display the contents of a buffer. Use the **Load Buff** command to load the IFF image you wish to display, then use the **Show Still** command. The **Buf** field must contain the number of the buffer to display.

### Play ANIM

The **Play ANIM** function does for animations what **Show Still** does for IFF files. ANIM files may be loaded and played directly from disk or played from a buffer. To load and play a file from disk, the **File** field must contain the name and location of the file.

To play an ANIM already loaded into a buffer, the **Buf** field must contain the number of the buffer to use. This is a useful feature for keeping a show moving smoothly. ANIM files can be very large and so take a lot of time to load. By loading an ANIM into a buffer at the beginning of a show or during a point where a picture is being displayed, you can create a smooth transition into the animation. This way there isn't a blank screen while the file is loading.

The **Play ANIM** feature doesn't make use of any of the transition effects before beginning an animation. It simply performs a **Flip** and begins the animation. If you wanted to **Dissolve** into the animation you would first load the ANIM into a buffer, then use **Show Still** to display the first frame of the ANIM as an IFF picture and perform the **Dissolve**, and finally, **Play ANIM** from the buffer. The final effect would be the animation dissolving onto the screen and then beginning to play. Here it is step by step:

#### To perform a transition into an animation:

- You will need 3 frames in the **Video Script Requestor**, so click the **Insert** button 3 times to create the clear frames. Select the first frame and then select the **Edit** button.
- In the **Frame Specs** requestor select the **Blank** button until it reads **Load Buff**.
- Change the **Buf** field to **1** (or any valid buffer number from 0 to 99).

- Select the **File** field and choose the ANIM file to play.
- Close the window. The action just created will load the animation into memory but not play it.
- Select the next frame and then the **Edit** button.
- Select the **Blank** button until it reads **Show Still**.
- Enter the buffer number that you loaded the ANIM file into above.
- Select any transition, such as **Dissolve**.
- Enter the number of seconds to pause before beginning the animation. Two seconds is usually right.
- Close the window.
- Select the final blank frame and then the **Edit** button.
- Select the **Blank** button until it reads **Play ANIM**.
- Change the value in the **Buf** field to the number of the buffer you loaded the animation into above.
- Enter the **Speed** and any **Repeat** values for the animation.
- Close the window. When played, the script just created will pause as it loads all of the files and then perform the chosen transition into the first frame of the animation before playing it.

#### Using MVP Transitions When Playing Anims:

When you play an ANIM, two transition buttons are active -- **Flip** and the **MVP** buttons. **Flip** indicates normal play of an ANIM. When a **MVP** transition is selected, the frames of the ANIM are shown one at a time with the indicated **MVP**

transition occurring between each frame. This makes possible many interesting effects, including text which scrolls through sections of the screen. For example, you could save a series of text screens from *VideoTitler* as an ANIM and then use the **MVP** transition to scroll the text on the display screen.

*Note: The only restriction when using **MVP** transitions with ANIM files is that the ANIM file must be recorded in severe overscan.*

#### Creating Scrolling Scripts using the MVP transitions:

Using **GrabANIM** in conjunction with any program which allows for severe overscan (or using the ANIM record feature in *Aegis VideoTitler*) provides a fast and easy way to create a scrolling script.

#### To create a scrolling ANIM:

- Open a new ANIM file using **GrabANIM** (or the ANIM record feature of *VideoTitler*).
- Make sure the program you are using is in **Severe Overscan** mode.
- Enter text and/or pictures, recording an ANIM frame between each — no movement is required.
- Close the ANIM file.
- In *Lights! Camera! Action!*, enter the following in the **Frame Specs** window.
- First, choose **Play Anim** in the **Blank** button.
- Select the path and filename of your ANIM.
- Enter the desired values in the **Repeat** and **Speed** edit fields.

*Note: The ANIM must have a two-frame overlap in order to repeat correctly.*

- Entering an ANIM Speed value will make the pauses between frames uniform.
- Select a Transition Speed of fast or slow.
- Select one of the MVP transitions.

*Note: To play a regular ANIM without transitions, select Flip.*

- Now you can play your scrolling script.

Each frame will appear in the order it was recorded.

## Play Score/Sound

This selection will not work until you have first loaded the score or sound using the **Load Score** or **Load Sound** command. When you want to play a score or a sound file, this selection will activate the buffer holding the music or sound file. See the section on loading scores for more information about loading a music or sound file into a buffer.

*Note: Scores and sounds must be in Sonix format.*

### Playing a score:

- Select **Blank** in the Frame Specs window and choose **Play Score/Sound**.

*Note: Time, buffer and the score/sound fields are the only options available with this frame selection.*

- Enter in the number of seconds you wish the score to play in the **Time** edit field.

*Note: If you do not enter a time, the music will play until the end of the script or the score, whichever comes first.*

- Enter in the **Buf** edit field the number you assigned to the score when you loaded it from disk. Be sure to use the proper buffer number.

There are a number of options available when playing a score or sound. You will not use the **Track** or **Pitch** options for scores as they have already been determined when the score was created in Sonix.

- Although you will usually want to play a score completely from beginning to end, as an option you may enter values in **T1** and **T2** to determine where to begin and end the playing of the score. **T1** determines how many 32nd notes from the beginning of the score to start playing and **T2** determines how many 32nd notes from the beginning of the score to end the music. For example, if the score is in common time (4/4) there are 32 32nd notes in each measure. Let's assume that you are dealing with a score that is only ten measures long. If you wanted to play only measures 2 thru 4 you would enter 33 as the value for **T1** and 128 for **T2**.

- You may also specify that a score (or a section of the score, see discussion of **T1** and **T2** above) repeats. Simply enter the number of times you would like to repeat the score in the **Repeat** field.

- The **Volume** option should not be used when playing a score. The **Volume** is determined when the score is created in *Sonix*. The **Volume** option *does* affect sounds. Its default setting of 0 produces full volume.

### Keying a visual event to a specific point in a music score

*Lights! Camera! Action!* gives you the ability to key visual events (**ShowStill**, **PlayANIM**, **SG Event**, **Color Bars**, **Blank**) to specific points in a musical score. You can control

when the visual event will occur by entering values into the **Frame** field in the **Frame Specs** requestor. The values in the **Frame** field describe the number of 32nd notes from the beginning of the playing score that you wish the visual event to occur. In 4/4 time there are 32 32nd notes per measure, so if you wanted a still picture to be displayed ten measures into a score you would enter 320 into the **Frame** field of the **Show Still** option. Here's how to do it step-by-step:

- In the **Frame Specs** requestor, use the **Load Buff** command to load a picture file into a buffer.
- In the next frame of your script, use the **Load Score** command to load a musical score into another buffer.
- In the next frame, use the **Play Score/Sound** command to begin playing your score.
- Now use the **Show Still** command (with the same buffer number you specified earlier) with a number entered into the **Frame** field to tell the program where to begin showing the still picture during the playing of your score. (Again, the value is the number of 32nd notes from the beginning of the score that you want the image to be displayed.)

#### Playing a sound:

*Note: Remember that you must load a sound into a buffer using the **Load Sound** command before you can play it.*

- Click the left mouse button on the word **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list of frame options and select, **Play Score/Sound**.

*Note: Time, buffer and the Score/Sound fields are the only options available with this frame selection.*

- Position the cursor in the **Time** edit field at the top of the **Frame Specs** requestor. Enter in the number of seconds you wish the sound to play.

*Note: If you do not enter a time, the sound will play until the end of the script or the sound whichever comes first.*

- Enter in the number of the buffer you assigned to the sound when you loaded it from disk. Be sure to use the proper buffer number.
- You may select one of 4 tracks (1-4) for each sound that plays. This means that you can load up to four sounds into four separate buffers and have them all play at once using four separate **Play Sound** commands. Tracks 1 and 2 are left channel. Tracks 3 and 4 are right channel.

*Note: If you do not specify a track (0), the next available track will be used. If you want to play a sound while a score is playing, check to see if the score uses all 4 tracks. If you assign a sound to play while a score is playing and the sound uses one of the score's assigned tracks, the score's instrument on that track will not play while the sound uses its specific track.*

- You will not use **T1** or **T2** while playing sounds *except* during **ANIM Sound Events**.
- You may raise or lower the **Pitch** of a sound. The values here are the same as for MIDI instruments (60=default). By entering a value of 61, you would raise the pitch of the note by one half-step. By entering 59, you would lower the note a half-step. Entering a value of 72 would raise the pitch one octave (12 half-steps), and so on.
- The **Volume** option lets you determine the loudness of a sound. The default setting of 0 provides the loudest possible output. Values from 1 to 255 may be entered to modify the volume of a sound with 1 being barely audible and 255 being the loudest.

## Pause

The **Pause** command causes the show to stop until the left mouse button is pressed. This works in any of the **Play** modes, including **Play Once** and **Play Loop**.

## Loop Point

**Loop Point** works in conjunction with the **Play Loop** play mode. It specifies the point in the show where *Lights! Camera! Action!* is to return once it reaches the end of the script. This is useful for displaying a series of introductory pictures or animations and then looping through the main body of the script.

**Loop Point** does not use any of the fields on the Frame Specification requestor such as **Time** or **Buffer**.

## Sound Event

If you want to play a sound during an ANIM style animation, you may assign a *Sonix* instrument file to a specific frame of an animation. Before you can use **Sound Event**, you must have an ANIM and a sound file loaded into separate buffers. See the section on loading buffers.

*Note: ANIMS can be played directly from disk but sounds must be loaded into buffers.*

- Insert a new frame below a frame that loads an ANIM into a buffer in the **Video Script** requestor.
- Select **Edit**.
- Click the mouse on **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list and select **Sound Event**.

- Enter in the buffer number for the sound file you wish to use.
- Set the **Frame** field to the frame in the ANIM you want to trigger the sound. For example, if you enter 15, the sound will play the first time the ANIM reaches frame 15. In repeating ANIMs, if you want the sound to repeat every time a particular frame is reached, use a negative symbol in front of the frame number (Eg. -15 will cause the sound to repeat with the ANIM).
- You may select from one of four tracks (1-4) for the sound to play on. If you do not specify a track (0), then the next available track will be used.

*Note: You may use **T1** as a "note release" trigger and **T2** to stop the sound completely. If you want the sound to begin decaying 5 frames after it begins, then indicate 5 next to **T1**. If you want the sound to completely stop 2 frames after that select 2 next to **T2**. This allows you to control in fine detail the sound during an animation.*

*Examples:*

- 1.) You have a repeating ANIM which is 10 frames long. You have entered a value of 5 in the **ANIM Repeat** field. This will cause the ANIM to repeat 5 times before *Lights! Camera! Action!* goes on to the next event in your script. The frame numbers in this example would then be numbered 1-10,1-10,1-10,1-10,1-10. If you want a sound effect to occur each time the ANIM repeats you will want to enter a *negative* number in the **Frame** field of your **Sound Event**. For example, entering a -3 will cause the sound to begin at the third frame of the repeating ANIM each time it plays. Entering a 4 in the **T1** field will cause the sound to begin decaying 4 frames later, at frame 7. Entering a 3 in the **T2** field will cause the sound to stop completely 3 frames after that, at frame number 10. Entering a 3 (positive) in the **Frame** field will cause the sound to occur only during the third frame of the ANIM the first time it is played. Entering a 42 would cause the sound event to occur only

during the second frame of the the 5th repeat of the ANIM (since each ANIM is 10 frames in length).

2.) You have a non-repeating ANIM which is 25 frames long. You want the music to begin as soon as the ANIM begins to play. So you enter a value of 0 in the frame field. This causes the sound to begin at the first frame of the ANIM. You then want the note to start decaying at frame 18. So you enter a value of 18 in the T1 field. You want the sound to stop completely 2 frames before the ANIM ends, so you enter a 5 in the T2 field. (The sounds begins to decay at frame 18 and stops completely 5 frames after that at frame 23.)

*Note: You may have multiple sound events occurring at the same time during ANIM playback. If you wanted four different sound events (the maximum number of tracks available), you would load four different buffers with sounds and create four separate sound events.*

- You may adjust the **Volume** in the same manner as when playing a sound.
- Insert a new frame directly below the **Sound Events** frame in the **Video Script** requestor.
- Click the mouse on **Blank** in the upper left corner of the **Frame Specs** window and hold the button down.
- Move down the list and select **Play ANIM**.
- Enter in the buffer number for the ANIM file you wish to use.

### SG Event

The **SG Event** command (under the **Blank** button in the **Frame Specs** requestor) is provided to give you control of the *SuperGen* genlock during the playing of ANIMS. Simply specify the frame number of the ANIM where you would like

your *SuperGen* settings to occur. For example, you could have a two second dissolve occur starting at frame 30 of a 50 frame ANIM.

### Using *Lights! Camera! Action!* with the *SuperGen*™ Genlock:

*Lights! Camera! Action!* includes complete software controls for the excellent *SuperGen* genlock device from Digital Creations. In order for the program to function with *SuperGen*, you must first install two library files (*Copinit.library* and *Supergen.library*) supplied with the *SuperGen* into the *Libs* directory of your *Lights! Camera! Action!* program disk (or onto whatever disk you are booting from as a Workbench disk before running the program). Instructions for doing this are in the manual which came with your *SuperGen*.

Once you have booted from a disk which has the *SuperGen* libraries installed and you run *Lights! Camera! Action!*, you will notice that the **SG** button in the lower right hand corner of the **Frame Specs** requestor is now active.

Most of the commands in *Lights! Camera! Action!* can have *SuperGen* values attached to them allowing you to control the mix between computer graphics and external video quite easily. After clicking on the **SG** button, you will see the **SG Dissolve Settings** window with three proportional gadgets. The top one is a two-dimensional **Dissolve Level** gadget which lets you control the mix between computer graphics and external video by moving around the small black gadget. To the right of this gadget are numeric readouts which let you know exactly which values will be passed to the *SuperGen*. (For more information about these values or other aspects of the *SuperGen*, please consult your *SuperGen* manual.) The two **Time Span** sliders at the bottom of the window give you control of the time that will be taken to move the settings from their current, existing settings to the ones you have specified in the **Dissolve Level** settings. These times are expressed in "jiffies" (one-

sixtieth of a second). Notice that different times can be used for computer graphics and external video for artistic effects.

*Note: Although the SuperGen is equipped with manual sliders, once you call out a SuperGen setting in your Lights! Camera! Action! script, the SuperGen will come under computer control and the sliders and other controls on the SuperGen will be inoperative until the entire script is completed.*

### Clear Events

Selecting **Clear Events** will cause all ongoing events to be cancelled. Sound effects, ANIM playback, and Sonix scores will all stop playing if **Clear Events** is selected.

### Time

The **Time** field tells *Lights! Camera! Action!* how long to perform the selected action. The value entered is in seconds and shows up under the **Time** column on the frame list.

### File

Selecting **File** will bring the storage requestor to the screen. From the storage requestor you select the IFF picture, ANIM, score or sound file to be used. *Lights! Camera! Action!* will load the file and path you select into the field for you. Some commands, such as **Pause**, do not make use of files, so these fields will be inactive.

### Buffer (Buf)

If the action uses a buffer, its number is entered in the **Buf** field. The **Clear Buffer**, **Load Buffer**, **Show Still**, **Play Score/Sound**, **Sound Event** and **Play ANIM** actions can all make use of this field. The number must be between 0 and 99.

### ANIM Repeat

This field works with the **Play ANIM** action and causes the file to loop. An animation will play the amount of loops you enter. Keep in mind when creating repeating ANIMs that the first 2 frames and the last two frames must be identical. (In a 20-frame ANIM, frame 1 must match frame 19 and frame 2 must match frame 20.)

### ANIM Speed

This field also is used in conjunction with **Play ANIM** and allows you to specify the amount of time that will elapse between ANIM frames. Values entered into this field are in "jiffies" (1/60th of a second). For example, entering a 1 would cause an additional 1/60th of a second delay between each frame with an overall speed result of 30 frames per second. The default value of 0 gives the top ANIM playing speed of 60 frames per second.

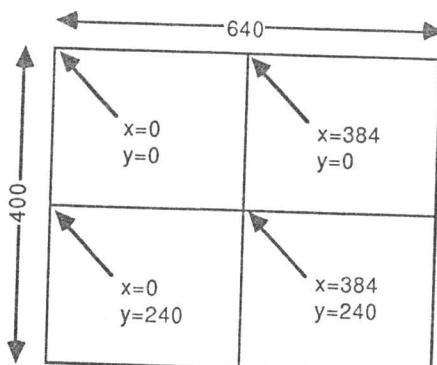
*Note: The number of ANIM frames which can be played per second depends upon the complexity of your animation. In detailed ANIMs, there may already be 4 or 5 jiffies occurring between each frame. In these cases, only values greater than 4 or 5 would have an effect.*

### Offset

Some paint programs are capable of creating pictures that are much larger than the boundaries of the screen. The **Offset** edit fields in the **Frame Specs** requestor are for specifying which region of a large picture will appear on the screen during a script. If the **x** and **y** numbers are zero, the upper left region of the picture will appear on the screen, extending as far as the screen boundaries allow.

Changing the **x** and **y** numbers specifies the upper left corner of a new region of the picture. The new upper left corner is found by shifting to the right **x** number of pixels and then shifting down **y** number of pixels. For example, let's say we have a picture created in lo-res, that is 640 pixels wide and 400 pixels high. Because this is twice as large as a normal lo-res screen (384x240), we could only show one-fourth of the picture at a time. The following diagram shows which values you would enter to show each region:

*Note: Lights! Camera! Action! automatically displays in Severe Overscan.*



*Note: These regions are arbitrary; you could show the center of the picture by entering **x=128** and **y=80**.*

To show a different region of a large picture:

- In the Frame Specs requestor, choose **Load Buff** in the **Blank** button.
- Select the path and filename of the big picture.

*Note: You must first load a large picture into a buffer before you can show it.*

- Enter a **Buf** number and close the frame.
- In a new frame, choose **Show Still** in the **Frame** button.
- Enter the **Buf** number of the big picture.
- Enter the coordinates of the desired region in the **x** and **y** **Offset** edit fields.
- Choose a transition and close the frame.
- Now you can play your script.

## Transitions

All of the transitions except **Fade** and **Flip** perform a gradual blend from the first color palette to the next. **Fade** performs a fade to black, changes the palette, and then fades into the next picture. **Flip** jumps between pictures and changes the palette instantly.

*Note: When two images with separate palettes are merged or otherwise manipulated, the palettes will switch once the image becomes the primary picture on the screen of the Amiga. Just prior to this taking effect, the new image may use the same palette as the prior picture, resulting in strange colors. This effect will last only a moment and shouldn't detract from your presentation.*

**Dissolve** performs a pixel-by-pixel fade from one picture to the next.

**Diagonals** and **Diamonds** are pattern wipes that occur throughout the screen during the transition.

The **Wipe** option has a sub-menu with nine choices. Each X in the sub-menu represents the direction the wipe will start from. Choosing the first X (upper left corner) produces a diagonal wipe moving from the upper left corner of the screen to the lower right. Choosing the second X down on the left causes a vertical wipe moving from left to right on the screen.

In the center of the **Wipe** sub-menu is a crosshatch symbol. This represents a wipe which begins in the center and moves to the edges of the screen.

The **Block** option also has a sub-menu, but with ten descriptive choices. Each choice performs its particular effect with small squares of screen area. Imagine the screen equally divided into 144 blocks. These are the blocks used during any block transition.

**Collapse** makes the picture scroll from the center of the screen toward the top and bottom.

**Center Scroll** scrolls the picture through a "window" in the center of the screen.

**Bottom Scroll** scrolls the picture through a "window" at the bottom of the screen.

The **MVP** label at the bottom stands for Multiple View Port. The **MVP** transitions and **Collapse**, **Center Scroll**, and **Bottom Scroll** split the screen at one or more points so that you can have the following on screen at once:

- Two different color palettes.
- Two different color cycling effects.

You will see a small border between frames, but you can minimize the effect by changing the picture's background color in a paint program.

There are 4 **MVP** transitions:

**Wipe** wipes the top picture away revealing the new picture underneath.

**Scroll** slides both the previous and new picture at the same time.

**Cover** covers the old picture with the new one.

**Reveal** slides the old picture revealing the new picture underneath.

Each of these **MVP** transitions can be controlled further by selecting **Up**, **Down**, **In**, **Out**, **Up 4** or **Down 4**. These options control the direction of the picture movement.

*Note: All transitions can be controlled by the Speed option.*

## Saving a Script

---

Once a script is complete, it may be saved to disk for later use.

*Note: Before saving the file, be sure the correct settings are chosen in the Display menu.*

To save a script to disk:

- Choose the desired option from the **Save** submenu in the **Project** menu. The Storage requestor will open.
- Open the root directory and then the subdirectory (if any) you'd like to put the file in.
- Click in the **file** edit field with the Selection button and type a name for the script file.
- Press **RETURN**. The drive will whir, the storage requestor will disappear, and a copy of your work will be recorded in the specified directory. Any further changes you make will not be recorded unless you save again.

You won't have to type the filename the next time you save — just choose it from the directory's list in the storage requestor and select OK.

## Playing Scripts

There are two different ways to play a script once it has been created. The currently active script in *Lights! Camera! Action!* may be played with any of the commands under the Video menu or you may use a separate module named *ShowLCA*.

*ShowLCA* is a program run from CLI. It automatically plays the requested script without going through the *Lights! Camera! Action!* program. *ShowLCA* may be distributed without copyright infringement. This way you can produce a script, place it on a disk with *ShowLCA* and give it away without giving away *Lights! Camera! Action!*

To play a script with *ShowLCA*:

- At the CLI prompt, type:

```
ShowLCA scriptname
```

Type the path and name of the script in place of *scriptname*. The script will play in Play Loop mode.

To change the playback mode of a script for *ShowLCA*:

- Open the script file, which is an ordinary text file, in a word processor or text editor.
- Replace the **Play Loop** command with the **Play Once** command. Remove the command altogether if you'd like the script to play back in Play Manual mode.

In *Lights! Camera! Action!* there are three options under the Video menu for playing a script: **Play Manual**, **Play Once**, and **Play Loop**.

### Play Manual

This mode steps through the script one frame at a time as the Selection mouse button is pressed. The Menu button will back the show up by one frame. In this way the mouse acts like a remote control on a slide projector for frame advance and frame reverse. The script will be presented one time only in this mode.

### Play Once

The **Play Once** mode plays the script one time automatically. If there is a **Pause** in the script *Lights! Camera! Action!* will wait for the Selection button to be pressed before advancing.

### Play Loop

**Play Loop** plays the script from beginning to end in a continuous loop. If there is a **Loop Point** in the script the looping will occur between the end and that point in the script. To interrupt a script in **Play Loop** press the Menu button.

To load a script for playing:

- Choose **Open Script** from the **Project** menu.
- From the storage requestor select the script you want to play.
- Select the **OK** button.

# CHAPTER

## 4

# DISPLAY OPTIONS

The **Display** menu is important in the building of a script. From here you specify whether the show will make use of halfbrite, color cycling, or interlace, and the resolution of the pictures. Once selected, these settings are saved with the script.

### To change the screen display:

- Pull down the **Display** menu.
- Make all your choices at once by clicking on them with the Selection button. An **ATTENTION!** requestor will appear, warning that the Bufffers will be erased.
- Select **OK** to proceed.

### NTSC/PAL

The Amiga can transmit an NTSC signal in order to be compatible with regular television and video equipment. This is a standard set by the National Television Systems Committee for broadcasting in the United States. Europe and Australia, on the

other hand, use the PAL format. One obvious difference between the two is that in Lo-res, the PAL display has an extra 60 scanlines (see Table 4.2).

*Lights! Camera! Action!* automatically displays in severe overscan. This means that when the script is recorded onto video, any hint of a screen border is eliminated. If your picture or non-scrolling ANIM was not created in overscan mode, it will still be correctly displayed. Also, overscan and non-overscan pictures may be used in the same script (as long as they are in the same resolution).

### NTSC

OverScan	Off	Medium	Severe
Lo-Res	320x200	352x220	384x240
Video-Res	320x400	352x440	384x480
Med-Res	640x200	704x220	768x240
Hi-Res	640x400	704x440	768x480

Table 4.1: NTSC Screen Display Options (columns x scanlines).

*Lights! Camera! Action!* determines which type of machine you have and automatically switches to that mode upon startup.

### PAL

OverScan	Off	Medium	Severe
Lo-Res	320x256	352x276	384x300
Video-Res	320x512	352x552	384x600
Med-Res	640x256	704x276	768x300
Hi-Res	640x512	704x552	768x600

Table 4.2 PAL Screen Display Options (columns x scanlines).

### Lo-res

This resolution requires the least amount of memory and is the default when you start *Lights! Camera! Action!*. It also enables the fastest ANIM speed and the use of halfbrite. A palette of 32 colors (64 with halfbrite) is available.

### Video-res

This doubles the vertical resolution of Lo-res, keeping the palette of 32 colors (64 with halfbrite).

The video "bandwidth" is defined by the number of colors available multiplied by the number of pixels (columns x rows). Video-res provides the highest bandwidth, making it the most functional for video recording. Granted, you could double the horizontal resolution by using Hi-res, but you would need more memory and your palette would be cut in half. Moreover, a higher horizontal resolution does not have much effect because of the poor resolution of the videotaping process.

## Med-res

This is compatible with most Amiga software, such as *Aegis Draw Plus* and *Aegis Images HR*. This is not recommended for video work — although it has the same number of pixels as **Video-res**, it only has half the colors. **Halfbrite** is not supported.

## Hi-res

This has the maximum number of pixels, providing the sharpest imagery; however, only 16 colors are available. The program functions, with the exception of a few, aren't any slower than for **Med-res**. Two megabytes of memory are recommended.

## Interlace

You'll notice that the display for either **Video-res** or **Hi-res** vibrates. This is because the video display is constantly being updated — like frames in a movie. Instead of having all 400 (with overscan: 440, 480) scanlines on the screen at once, every other scanline is sent to the screen every 1/60th of a second. This technique is called "interlace." The timing makes the display more compatible with video recording.

You can also turn on **Interlace** for **Lo-res** and **Med-res** displays. For example, if you have an *Aegis ArtPak* picture created in **Lo-res**, **Interlace** will send each line of pixels to the screen twice. The result is that the dark lines in between will be filled, making the display look better on video.

## HalfBrite

The Amiga 500, 2000, and early models of the 1000 have the "halfbrite" capability. When you turn on **Halfbrite**, another set

of colors is added to the color palette. These are duplicates of the original colors, only they're half as bright — thus the term "halfbrite." **Halfbrite** is available for **Lo-Res** and **Video-Res** modes only. To enable **Halfbrite** mode, choose **Halfbrite/HAM** from the **Video** Menu.

*Note: If your Amiga 1000 doesn't support halfbrite, see your dealer about ordering the version of the Denise chip which has halfbrite, chip number #8362, revision 6 or later.*

## Hold and Modify

The **Hold and Modify** (commonly called "HAM") menu option allows *Lights! Camera! Action!* to display both picture files and ANIMs created by other programs in HAM mode. HAM mode enables the Amiga to display 4,096 colors on the screen at once. HAM is available for **Lo-Res** and **Video-Res** modes only. To enable HAM mode, choose the **HalfBrite/HAM** option under the display menu.

## Cycling

Color cycling is a feature found in most paint programs for the Amiga. It allows you to cycle the colors through a range of paint pots in the palette. This can be used to make a limited form of animation. If any of the IFF pictures you are using in the show make use of color cycling, you must make sure this option is turned on.

## The Attention Requestor

You can bring up the **ATTENTION!** requestor by pressing the **HELP** key. This will show the amount of available memory and the width, height, and depth of the screen, in that order. Screen depth is measured in bit planes. The number of available colors

## LIGHTS! CAMERA! ACTION!

is a function of the number of bit planes — halfbright and HAM adds a sixth bit plane, thus more colors.

## APPENDIX A

### THE STORAGE REQUESTOR

*Lights! Camera! Action!* uses a standard storage requestor for retrieving files (IFF picture files, sound and music data, and scripts) or saving them to disk (or memory).

*Note: For complete details on the operation of your Amiga, refer to the manuals included with your Amiga.*

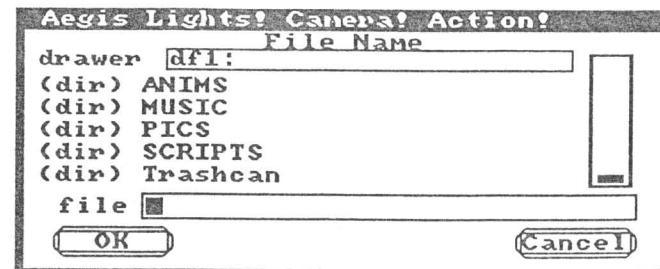


Figure 3: The Lights! Camera! Action! Storage Requestor.

The **drawer** edit field shows the current root directory and any open subdirectories, sub-subdirectories, and so on. (Subdirectories are called "drawers" if they have an associated icon (symbol) on the Workbench screen.) For example:

- **DF0:** — The open root directory is the disk in the internal drive.
- **RAM:** — The open root directory is the Amiga's random access memory (called RAM), which for all intents and purposes, can act as a disk. This is handy if you're continually accessing the same file and you don't want to wait for the mechanical drive to read it every time; you would just save it to RAM: and instantly access it from there. It's also useful as a temporary holding area if you don't have enough disk space. Just remember that everything in RAM: is erased when you either turn off or reboot the computer.
- **DF1:pics** — The open root directory is the disk in the external drive and the open subdirectory is pics (where the IFF picture files are kept).
- **DH0:Lights! Camera! Action!/pics** — The open root directory is the hard drive, the open subdirectory is Lights! Camera! Action!, and the open sub-subdirectory is pics.

To open a directory and choose a file:

- Choose **Open Script** from the **Project** menu. This brings up the storage requestor.
- Click in the **drawer** edit field with the Selection button. A cursor will appear.
- Use the **BACK SPACE** and **DEL** keys to erase any current information.
- Type the root directory, colon, and the subdirectory — without spaces. For subdirectories, and so on, separate them with forward slashes. If you don't know what subdirectories are

contained in that root directory, don't type anything after the colon and you'll see a listing after you complete the next step.

- Press the **RETURN** key. The disk drive will whir and the list will change to show the contents of the current directory.
- To scroll through the list, drag the scroll bar on the right of the requestor.
- To choose an item from the list, click on the name with the Selection button. If it's a directory, it will appear in the **drawer** edit field. If it's a file, it will appear in the **file** edit field.
- To go back to the root directory so you can open a new subdirectory, click in the **drawer** edit field, erase everything after the colon, and press **RETURN**. The list will change to show the contents of the root directory.

When you're done looking at the requestor, select its **Cancel** button to close it.

You may need several data disks while working with *Lights! Camera! Action!* The storage requestor will let you swap disks as needed.

To use more than one data disk with a drive:

*Note: Never remove a disk while the drive light is still on.*

- After you've called up the storage requestor, remove the original data disk from the external drive and insert the data disk you want to access.

*Note: The program is preset to look for certain directories which may not be on your new data disk. If so, a requestor saying **Wrong Diskette?** will appear. Select **OK** and go on to the next step.*

- Click in the **drawer** edit field with the Selection button and use the BACK SPACE and DEL keys to erase the old information.
- Enter **DF1:** and press RETURN. The disk drive will whir and the list will change to show the contents of the new disk.

## APPENDIX B

### USING GRABANIM

**GrabANIM** permits sequences of graphics screens from any Amiga program to be captured and saved onto disk in the Aegis/Sparta ANIM format. You can then include these ANIM files in *Lights! Camera! Action!* presentations.

For example, you could put some text on the screen using *Deluxe Paint*™, make a brush out of it, save the screen using **GrabANIM**, move the brush a little, save again, etc. The end result would be an ANIM file which, when played back would show your text moving around the screen.

**GrabANIM** is designed to run as a background task on the Amiga and to "grab" graphic screens and add them to an open ANIM file on disk. **GrabANIM** always grabs the "highest" or visible screen on the Amiga display. To use the program follow these step-by-step instructions:

- 1.) Open a CLI window by double clicking on the CLI icon.
- 2.) At the **1>** prompt type **Grabanim**
- 3.) Now the type of input device can be selected:

**J** - Allows a joystick in the second mouse port to be used for frame entry. Hitting the fire button will add a frame to the ANIM file. Pushing the joystick up (forward) will close the file.

**M** - Allows a mouse in the second mouse port to be used for frame entry. Pressing the left button will add a frame and selecting the right button will close the ANIM file.

**K** - Allows a Koala Pad to be used for frame entry. Button selections are the same as for mice.

You may also use the keyboard although it should not be used if multi-tasking with another program which uses the keyboard. Keyboard controls are:

**O** - Open ANIM file **A** - Add frame to ANIM File **C** - Close ANIM file

Three additional controls are available:

**P** - This selection turns off the multi-colored band which appears across the top of the screen after a frame is recorded. This multi-colored band may cause problems with some programs

**\$** - This control will make the active window the recording window even when it is covered by another window.

**Q** - Quit to the CLI prompt.

Note: **J**, **M**, **K**, and **O** will advance to step 4. **A**, **C**, **P**, and **\$** will return to step 3.

- 4.) Next the path and file name can be entered.

Example: Enter file name (blank to abort):

**DF1: Test.ANIM**

- Now the ANIM can be created using the input device selected in step 3 (Or hit ESC, then Q to return to the CLI).

Once your animation is finished close the ANIM file. Now you can use *Lights! Camera! Action!* or any other ANIM playback program to display your animation

## INDEX

color cycling	x,45,49
Cover	41
Delete	17
Diagonals	39
Diamonds	39
directories	52
Display	6,41,45
Dissolve	39
drag	3
drawer	52
Edit	6,8,15,18,25
End Black	8
Fade	39
File	36
Flip	39
Form Buffer	21
Frame	15,33
Frame Specs	8,15 18,19
frame, creating	8
genlock	34,35
GrabANIM	ix,21 27,55
Halfbrite	45,48
HAM	see Hold and Modify
Help key	49
Hi-res	47,48
Hold and Modify	... x,49
image, displaying	11
Insert	8,12,17,25
instruments	22
assign directory	9,23
Interlace	45,48
jiffies	37
joystick	55
Lo-res	47
Load Buf	20,24
Load Score	9,22,23,28

## LIGHTS! CAMERA! ACTION!

Load Sound ..... 9,24,28  
Loop Point ..... 32  
Loud Sound ..... 23  
Med-res ..... 48  
memory, see buffer  
Menu Bar ..... 3,5,6  
mouse ..... 3  
Multi-tasking ..... 2  
MVP ..... 27,40  
New ..... 8  
Next ..... 13,16  
NTSC ..... 45  
Offset ..... 37  
Open Script ..... 6,43  
overscan ..... 5,38,46  
    severe ..... 27  
PAL ..... 45  
palette ..... 39,47,49  
Pause ..... 32,36  
picture ..... 15  
    displaying ..... 11  
    oversized ..... 38  
Pitch ..... 31  
Play ..... 6  
Play ANIM ..... 25,34  
Play Loop ... 7,32,42,43  
Play Manual ..... 42,43  
Play Once... 7,14,42,43  
Play Score/Sound . 22,28  
PlayANIM ..... 21  
Prev ..... 16  
Project ..... 6,8  
Quit ..... 6  
RAM ..... 9,22,52  
Repeat ..... 29  
Reveal ..... 41  
Save ..... 41  
Score  
    loading ..... 10,23

    playing ..... 10  
script  
    saving ..... 41  
    playing ..... 42  
Scroll ..... 41  
Scroll bar ..... 53  
select ..... 3  
SG button ..... 34,35  
Show Still ..... 24  
ShowLCA ..... viii,42  
Sleep ..... 6  
SMUS ..... ix  
Sonix ..... ix,8,11  
    22,24,28  
Sound Event ..... 31,32  
sound  
    playing ..... 30  
    loading ..... 24  
Speed ..... 41  
Start Black ..... 8  
Storage requestor ..... 51  
SuperGen ..... 34,35  
synchronization ..... 29  
T1 ..... 29,31,33  
T2 ..... 29,31,33  
Time ..... 11,15,24,31,36  
Top ..... 16  
Track ..... 11  
tracks ..... 33,34  
transitions ..... 39  
Video ..... 6,7,8  
Video Script ..... 8,15  
    Figure ..... 16  
Video-res ..... 47  
VideoTitler ..... 27  
Volume ..... 29,31,34  
Wipe ..... 40,41  
Wrong Diskette ..... 53